Network Working Group
RFC: 304
NIC: 9077

D. B. McKay, IBM
February 17, 1972

Posted

9077

Rcvd 3Mar72

Catagories D3, D4, D7
Obsoletes: none
Updates: none

A Data Management System Proposal for the ARPA Network

Introduction

This proposal is being written to facilitate discussions on a design for a network Data Management System. It is not intended to be a complete and exhaustive design for the ultimate protocol allowing users to share data easily, but a frame work that will allow us to recognize and develop the necessary tools in a unified manner enabling the network to manage its resources to the best advantage of the user.

The fundamental intent here is not to try and solve an impossible problem, but to bring a necessary service capability to the user that will enable him to carry out applications that hitherto he has not been able to do. The intent is to be consistent with every other major function that has been developed in the network, i.e., NCP - 2nd level protocol, Telnet, and the Form Machine. The Data Management Service or Data Control Facility (DCF) will do the same thing only on a high level of application building on those tools that have already been developed in the network.

Data that is referred to and transmitted in this System will be considered a special class of data that is called network data. That is, it is named and characterized through a network datalanguage and all pertinent information as to where it can be located and what its structure is is kept in a

network catalog. Access to the data for its actual transmission will be done through NCP socket addressable routines in a manner similar to the way in which the SMFS at Santa Barbara works. It is feasible that the SMFS will become an active resource utilized by the DCF.

System Overview

There are six functionally and logically distinct areas that are identifiable in the Network Data Service (Figure 1), with subfunctions that can be categorized and discussed.

1. The user interface to the DCF. In an interactive environment such as the ARPA network, this interface would be serviced by Telnet supporting the local user at his terminal directing his request to the DCF. The DCF in this case would be a specialized server task.
2. The DCF or that functional unit responsible for coordinating all the activity of the Network Data Service. It also houses the interfaces to all other functions.
3. The Network Catalog or Directory which contains all information about network data.
4. The Data Reconfiguration Service or Form Machine that would be called on when data translation or reconfiguration is needed. This would be invoked automatically, when possible, by the DCF and would remove this responsibility from the user. For more specialize translation, however, the user will still be able to write programs for, and execute them on, the Form Machine.

5. The remote DCF or DCF' would contain enough function to recognize the request being made of it by the DCF It would be a server task to the DCF.

6. File xfer protocol would be a function that the DCF and the DCF' would initiate as the means to control data transfer in the network.

A more detailed discussion of each of these areas appears in the following sections.

User Interface

It was stated in RFC146 that the DCF should handle all network resources as a single resource and utilize it as best it can. This statement was also meant to incorporate the Data Computer and Unicon storage as part of this resource. The extent to which this can be done is an open question but the use of the Data Language developed by CCA would provide a consistent interface to the user utilizing these network services and possibly facilitate the use of the Data Compiler by the DCF.

It should be pointed out at this time that the DCF is a logical function that can reside anywhere including on the Data Computer.

The user should be allowed to enter all command and updates interactively to the DCF. The DCF will be a serving user process that will interface to the Telnet Server routine. The actual data of the terminal transmissions will be the commands and data the user will be transmitting to the

DCF. By adopting the Telnet protocol as the initial user interface, the DCF can be accessed by all the users with Telnet.

The actual user commands and data itself is an area that requires more investigation. The following comments offer suggestions as to what a final data language and manipulation language should do.

There are at least two logically distinct functions that must be performed. The actual defining and redefining of the data, and the request for service such as catalog entry and request for information. This proposal is not intended to provide access to every data base in the network; instead it is aimed at those files that are catalogued and known to the DCF in the manner analogous to the Data Computer's knowledge of its Data Base.

The following Data Description Concepts described in detail in CCA's Datalanguage are also useful in the DCF. First of all, the Data Containers are groups of nested boxes. The box represents the data or other data containers that are kept in the box. It represents a named set of locations in some storage medium. There are also several types of data container such as STRING, INT, REAL, PTR, ARRAY, LIST, STRUCT, and MIX. Finally, each of these containers can be named. The name can also qualify the item of interest by a concatenation of names to reflect the logical nesting of the containers.

Although storage and retrieval mechanisms should be the same as those proposed in the DL, initially it should not be necessary to implement all

DDT. By mapping the values produced as the initial name qualifiers, the DDT can be accessed by all the users plus values.

The actual data commands and data itself is an area that needs more investigation. The following comments offer suggestions as to what a final data language and manipulation language should be.

There are at least two logically distinct functions that must be performed. The actual defining and redefining of the data, and the request for services such as catalog entry and request for information. This proposal is not intended to provide access to every data base in the network. Instead it is aimed at those files that are catalogued and known to the DDT in the manner analogous to the Data Computer's knowledge of its Data Base.

The following data description concepts described in detail in CCA's Data-language are also useful in the DDT. First of all, the data descriptions are groups of nested items. The key represents the data or other data containers that are kept in the box. It represents a named set of locations in some storage medium. There are also several types of data container such as STRING, BIT, BOOL, FIX, ARRAY, LIST, STRUCT, and BIT. Finally, each of these containers can be named. The names can also qualify the item of interest by a concatenation of names to reflect the logical nesting of the containers.

Although storage and retrieval mechanisms should be the same as those proposed in the DC, initially it should not be necessary to implement all the functions that filter and manipulate the data. For example, in an initial implementation of a DCF it would not be necessary to provide the user with relational, boolean and computational operators. Users specifically interested in this type of service could be directed to use the Data Computer under his own initiative, or as a service of the DCF.

Several of the operators specified in the language are important and must be considered in the DCF.

The following list represents key operators for a DCF with a brief description of what each function is. For a more detailed discussion of each statement the reader should read the Datalanguage report by CCA.

The assign operation places values into the containers with the containers being single items or referencing other containers.

Subscripting allows selection by element number. It is a powerful tool for specifying, in large containers of data, the reference and transmissions of only the necessary parts. Files can be subsetted by containerization referencing fields and records which can be further classified by subscripting. This subscripting function can be further extended to include the DL's virtual list concept.

Maintenance of the files must be provided with the delete and add function applied to the container referenced data.

A second consideration in a catalog is the question of how feasible it would be to keep back-up or duplicate copies of network files. This of course raises the question of a multiple copy update protocol. I feel the discussion and development of this protocol, although important, can be postponed in lieu of keeping multiple copies of files that are primarily read-only files.

For experimental reasons the DCF should have at least one data base that is kept at different locations - possibly NIC, with the capability of access anyone of them in the event of system failure at other locations. This is an important point, it exploits one of the major advantages of a computer network namely more reliable data accessability.

Finally, the actual location of the network directory is an interesting question. In the interest of reliability it should be kept at multiple locations. The network directory can be logically separated into two segments. The local directory and distributed directory. Both parts refer to network data. The local segment is kept up-to-date relative to the network data that resides on that system. The network segment records the location of files that are duplicated on other systems and system pointers to references made of remote single system files.

Updates can be made to the network segment on a periodic basis. These updates will reflect changes in the local segments. If we consider "read-only" files distributed initially and local segments reflecting the changes in local files, the need for simultaneous update of multiple copies and

network segments of the catalog becomes much less critical. Based on the two segment approach to the network directory it seems most convenient to keep copies on all systems that have localized network data. This would include a catalog on the Data Computer.

Data Conversion (Form Machine)

The Form Machine represents an essential network function that can be invoked by the DCF when necessary. The Form Machine would be used in the same manner it was intended to be used only now the DCF would intervene in place of the user. This would represent a common interface to the network for the user. Having the user use the Datalanguage for file transfer and conversions would mean the DCF's management of his Form Machine Services.

The motivation behind the Form machine is consistent with a service that should be provided in a network data sharing facility, namely, application programs require different formats from program to program and the network should adapt to the individual program requirements. This is also true of console configurations and machine dependent data.

The modus operandi of the service is descriptions of data are supplied by the application programmer in forms that the service stores by name. In the case discussed here the DCF would invoke the data transformation on the network data stream by calling the forms by name. These would be a standard set of forms for machine dependent data that would be written as

part of a general implementation of the DCF and would be invoked when necessary by the DCF.

There are three conceptual connections to the Data Reconfiguration Service (DRS).

1. The critical connection between the orginating user and the DRS. In this case it would be the DCF. This raises the question of how the user would communicate with the Form Machine. He could use normal procedures and tie directly to the DRS by Telnet or he could allow the DCF establish a connection for him with his defined forms cataloged in the network catalog automatically.

2.-3. The other two connections are between the user process and the serving process which are made by the DRS through the NCP.

Since the Form Machine represents an invaluable service to the network it is imperative that it reside in several locations with user named forms available at each DRS location. This will ensure available service when needed. However, having the DRS invoked by the DCF raises two interesting areas of investigation.

The first is the question of the common interface that the network presents to the user. If the Datalanguage is to be the common interface, then is it practical and feasible for a mapping service to be performed by the DCF

that will convert Datalanguage statement into the proper parameters and statements necessary to the form machine? This is an area that has to be discussed and further investigated. I would encourage anyone to submit an RFC on this subject.

The second question is a simple one and has to do with the question of the file transfer protocol and whether or not it is sufficient to handle the requirements for the form machine connections from a remote location.

File Transfer Protocol (FTP)

The objectives of the FTP are to promote sharing of files, encourage implicit (without explicit login) use of computers, and to shield the user from variations in file and storage systems of various hosts. The use of these related operation in the FTP by the DCF allows us to extend these ideas to the restructuring and subfile level that was discussed in the user interface section.

In addition, request can be made to any DCF on any system to retrieve a file and transfer it to any other system. This means not only that implicit use of systems be encouraged but the user will be removed from the burden of explicitly having his system linked to the system transfering the file. For example, if someone is running an analysis program at BBN which may also have a DCF. A request for a file to be shipped to BBN from the Santa Barbara 75 could be made to the DCF at BBN without the user having to communicate with the 75 directly.

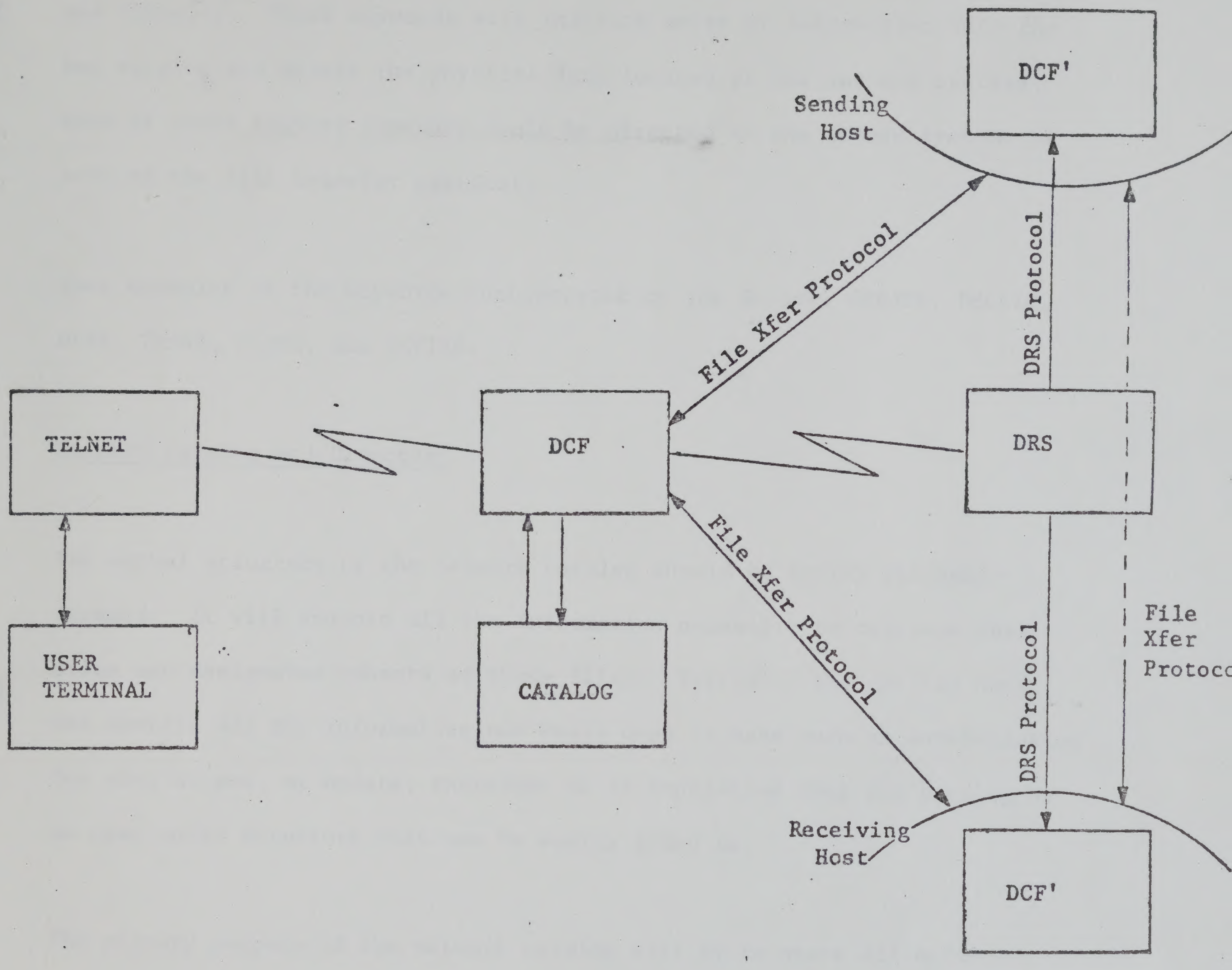
The sending and receive connection would be initiated by the DCF with the logical link between the two systems obeying the FTP. The only modification I can see in the FTP that would be necessary is an acknowledgement to the commands sent to the sending and receiving sites by the DCF. In addition, an acknowledgement to the end of file indication would be sent to the sending system and to the DCF. The rename from, rename to, delete and list request would be transmitted by the DCF directly with all acknowledgements being returned to it.

The remote DCF and DCF' mentioned earlier would recognize and handle all the FTP messages. In addition, it would recognize requests being made for a particular container or subset of the data. It should be able to recognize the information given to it, access the data requested and be able to strip off the necessary information requested and transmit it.

The complexity of the DCF' would depend on the amount of functional capability that was incorporated into the network portion of the Datalanguage.

Conclusion

This paper is intended to promote ideas and discussion in all of the areas mentioned. The principle outcome is to start a coordinated specification and implementation effort to provide data sharing in the network.



Network Data Service
System Overview

Figure 1

The keyword statements of the language are important for data manipulation and transfer. These keywords will initiate entry of information into the net catalog and access the physical data located at the various systems. Most of these keyword commands would be directed to the remote systems as part of the file transfer protocol.

Some examples of the keywords incorporated by the DL are, CREATE, DELETE, OPEN, TRANS, CLOSE, and DEFINE.

Network Catalog and Directory

The actual structure of the network catalog should be fairly straightforward. It will contain all the information necessary to retrieve data files and designated subsets of those files. Initially the catalog need not contain all the information one would hope to have such as authorization for use, access, or update, therefore it is imperative that the catalog be an open ended structure that can be easily added to.

The primary purpose of the network catalog will be to store all network data file structure information that the user has entered via the Datalanguage. It must also contain an indication of how the users logical description of his file is associated with the actual physical file and location. This physical information must contain the proper pointers and addresses to actually retrieve the data. Since the class of files we will be dealing with are network files that will be accessable by the network user function such as in the SMFS, the addressing information can be pathnames as suggested in the Rename Convention in the file transfer protocol.

The keyword statements of the language are important for data manipulation and transfer. These keywords will initiate entry of information into the net catalog and access the required data located at the various systems. Most of these keyword commands would be directed to the remote systems as part of the file transfer protocol.

Some examples of the keywords incorporated by the [illegible] are, CREATE, DELETE, OPEN, READ, CLOSE, and WRITE.

Network Catalog and Directory

The actual structure of the network catalog should be fairly straightforward. It will contain all the information necessary to retrieve data files and designated subsets of those files. Initially the catalog need not contain all the information one would hope to have such as authorization for use, access, or update, therefore it is imperative that the catalog be an open ended structure that can be easily added to.

The primary purpose of the network catalog will be to store all network data file structure information that the user has defined via the data language. It must also contain an indication of how the users logical description of his file is associated with the actual physical data and location. This physical information must contain the proper pointers and addresses to actually retrieve the data. Since the class of files we will be dealing with are network files that will be accessible by the network data functions such as in the [illegible], the addressing information can be patterned as suggested in the [illegible] Convention on net file transfer protocol.